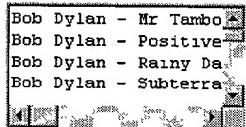


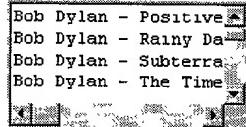
100

FIG. 1



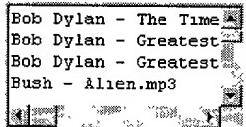
210

User starts at some point in the list



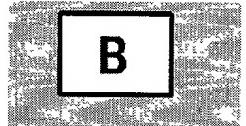
220

Scrolling starts off slow (one line per second)



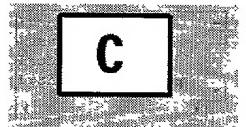
230

But soon speeds, so it is difficult to read the entries as they pass by



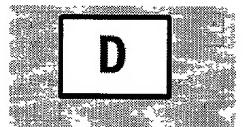
240

Large "helper" letter replaces list



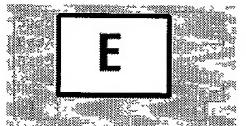
241

This now increments at a rate of approximately 1/second...



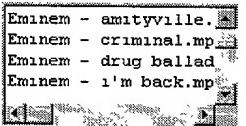
242

as long as the user holds down the scroll button



243

When the user reaches the vicinity of interest they release the scroll button



244

The list of entries returns to the screen

FIG. 2

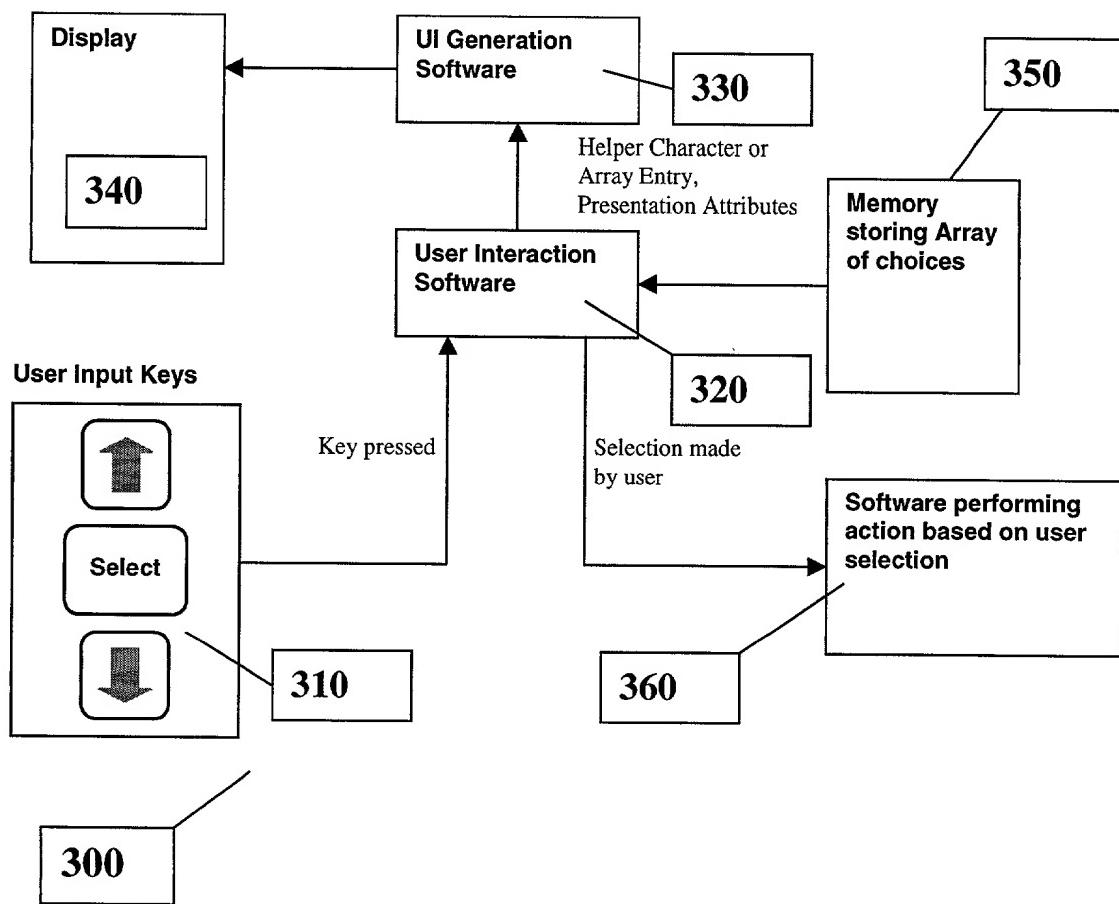


FIG. 3

```

*** Pseudo C Code ***
if (ScrollButtonPressed)
{
    if (ButtonPressed == BUTTON_UP)
    {
        DirectionOfScroll = SCROLL_UP;
    }
    else
    {
        DirectionOfScroll = SCROLL_DOWN;
    }
    if (ButtonPressDuration < T1)
    {
        ScrollStyle = SLOW;
    }
    else if (ButtonPressDuration >= T1)
    {
        ScrollStyle = FAST;
    }
    else if (ButtonPressDuration >= T2)
    {
        ScrollStyle = HELPER;
    }
}
else // Scroll Button Not Pressed
{
    ScrollStyle = NONE;
}
if (ScrollStyle == SLOW)
{
    wait(SlowScrollDelayLength);
    MovePointerInArray(DirectionOfScroll);
    DisplayNewArrayEntry;
}
if (ScrollStyle == FAST)
{
    wait(FastScrollDelayLength);
    MovePointerInArray(DirectionOfScroll); ← 420
    DisplayNewArrayEntry;
}
if (ScrollStyle == HELPER)
{
    wait(HelperScrollDelayLength);
    ChangeHelperCharacter(DirectionOfScroll); ← 430
    DisplayNewHelperCharacter; ← 410
}
*** Pseudo C Code End ***

```

400

420

430

410

FIG. 4